

Cryptanalysis of Two McEliece Cryptosystems Based on Quasi-Cyclic Codes

Ayoub Otmani, Jean-Pierre Tillich and Léonard Dallot

Abstract. We cryptanalyse here two variants of the McEliece cryptosystem based on quasi-cyclic codes. Both aim at reducing the key size by restricting the public and secret generator matrices to be in quasi-cyclic form. The first variant considers subcodes of a primitive BCH code. The aforementioned constraint on the public and secret keys implies to choose very structured permutations. We prove that this variant is not secure by producing many linear equations that the entries of the secret permutation matrix have to satisfy by using the fact that the secret code is a subcode of a known BCH code. This attack has been implemented and in all experiments we have performed the solution space of the linear system was of dimension one and revealed the permutation matrix.

The other variant uses quasi-cyclic low density parity-check codes. This scheme was devised to be immune against general attacks working for McEliece type cryptosystems based on low density parity-check codes by choosing in the McEliece scheme more general one-to-one mappings than permutation matrices. We suggest here a structural attack exploiting the quasi-cyclic structure of the code and a certain weakness in the choice of the linear transformations that hide the generator matrix of the code. This cryptanalysis adopts a polynomial-oriented approach and basically consists in searching for two polynomials of low weight such that their product is a public polynomial. Our analysis shows that with high probability a parity-check matrix of a punctured version of the secret code can be recovered with time complexity $O(n^3)$ where n is the length of the considered code. The complete reconstruction of the secret parity-check matrix of the quasi-cyclic low density parity-check codes requires the search of codewords of low weight which can be done with about 2^{37} operations for the specific parameters proposed.

Keywords. McEliece cryptosystem, quasi-cyclic codes, BCH codes, LDPC codes, cryptanalysis.

1. Introduction

Since the introduction of the McEliece public-key cryptosystem [17], several attempts have been made to propose alternatives to the classical Goppa codes. The main motivation is to drastically reduce the size of the public and private keys, which is of real concern for any concrete deployment. For instance, the parameters suggested in the original cryptosystem, and now outdated, are about 500 Kbits for the public key and 300 Kbits for the private key. The reason of such a large amount comes from the fact that McEliece proposed to use as public key a generator matrix of a linear block code. He suggested to take a code that admits an efficient decoding algorithm capable to correct up to a certain number of errors, and then to hide its structure by applying two secret linear transformations: a scrambling transformation that sends the chosen generator matrix to another one, and a permutation matrix that reorders the coordinates. The resulting matrix is then the public key. The private key consists in the two secret transformations and the decoding algorithm.

Niederreiter also invented [19] a code-based asymmetric cryptosystem by choosing to describe codes through a parity-check matrix. These two systems are equivalent in terms of security [16]. Their security relies on two difficult problems: the One-Wayness against Chosen-Plaintext Attack (OW-CPA) thanks to the difficulty of decoding large random linear block codes, and the difficulty of guessing the decoding algorithm from a hidden generator matrix. It is worthwhile mentioning that the OW-CPA character is well established as long as appropriate parameters are taken. This is due to two facts: first it is proven in [2] that decoding a random linear code is NP-Hard, and second the best known algorithms [8, 3] and [20, Volume I, Chapter 7] operate exponentially with the length n of the underlying code (see [10] for more details). However, the second criteria is not always verified by any class of codes that has a decoding algorithm. For instance, Sidel'nikov and Shestakov proved in [22] that the structure of Generalised Reed-Solomon codes of length n can be recovered in $O(n^3)$ (See for instance [24, page 39]). Sendrier proved [27] that the permutation transformation can be extracted for concatenated codes. Minder and Shokrollahi presented in [18] a structural attack that creates a private key against a cryptosystem based on Reed-Muller codes [21].

However, despite these attacks on these variants of the McEliece cryptosystem, the original scheme still remains resistant to any structural attack. Additionally, the McEliece system and its Niederreiter homologue display better encryption and decryption complexity than any other competing asymmetric schemes like RSA. Unfortunately, they suffer from the same drawback namely, they need very large key sizes as previously pointed out. It is therefore crucial to find a method to reduce the representation of a linear code as well as the matrices of the linear transformations.

A possible solution is to take very sparse matrices. This idea has been applied in [5] which examined the implications of using Low Density Parity-Check (LDPC)

codes. The authors showed that taking sparse matrices for the linear transformations is not a secure solution. Indeed, it is possible to recover the secret code from the public parity-check matrix. Another idea due to [25] is to take subcodes of an optimal code such as Generalized Reed-Solomon codes in order to decrease the code rate. But a great care has to be taken in the choice of parameters because in [26] it has been proved that some parameters are not secure. A recent trend appeared in code-based public key cryptosystems that tries to use quasi-cyclic codes [11, 1, 13, 12, 9]. This particular family of codes offers the advantage of having a very simple and compact description. Many codewords can simply be obtained by considering cyclic shifts of a sole codeword. Exploiting this fact leads to much smaller public and private keys. Currently there exist two public-key cryptosystems based upon quasi-cyclic codes. The first proposal [11] uses subcodes of a primitive BCH cyclic code. The size of the public key for this cryptosystem is about 20Kbits. The other one [1] tries to combine these two positive aspects by requiring quasi-cyclic LDPC codes. It also avoids trivial attacks against McEliece type cryptosystems based on LDPC codes by using in the secret key a more general kind of invertible matrix instead of a permutation matrix. For this particular system, the authors propose a public key size that is about 48Kbits.

In this work, we cryptanalyse these two cryptosystems. We show that the cryptosystem of [11] is not secure because it is possible to recover the secret permutation that is supposed to hide the structure of the secret quasi-cyclic code. We prove it by producing many linear equations that the entries of the secret permutation matrix have to satisfy by using the fact that the secret code is a subcode of a known BCH code. This attack has been implemented and in all experiments we have performed the solution space of the linear system was of dimension one and revealed the permutation matrix.

In a second part, we also suggest a structural attack of [1] exploiting the quasi-cyclic structure of the code and a certain weakness in the choice of the linear transformations that hide the generator matrix of the code. This cryptanalysis adopts a polynomial-oriented approach and basically consists in searching for two polynomials of low weight such that their product is a public polynomial. Our analysis shows that with high probability a parity-check matrix of a punctured version of the secret code can be recovered with time complexity $O(n^3)$ where n is the length of the considered code. An implementation shows that this recovery can be done in about 140 seconds on a PC. The final step that consists in completely reconstructing the original parity-check matrix of the secret quasi-cyclic low density parity-check code requires the search for low weight codewords which can be done with about 2^{37} operations for the specific parameters proposed.

The rest of this paper is organised as follows. In Section 2, we recall definitions and basic properties of circulant matrices. Section 3 gives a description of how to totally break the McEliece variant proposed in [11]. In Section 4 we propose a method to totally cryptanalyse the scheme of [1]. Section 5 concludes the paper.

2. Notation and Definitions

2.1. Circulant Matrices

Let \mathbb{F}_2 be the finite field with two elements and denote by $\mathbb{F}_2[x]$ the set of univariate polynomials with coefficients in \mathbb{F}_2 . Any p -bit vector $\mathbf{v} = (v_0, \dots, v_{p-1})$ is identified to the polynomial $\mathbf{v}(x) = v_0 + \dots + v_{p-1}x^{p-1}$. The *support* of a vector (or a polynomial) \mathbf{v} is the set of positions i such that v_i is non-zero and the *weight* $\text{wt}(\mathbf{v})$ of \mathbf{v} is the cardinality of its support. The *intersection* polynomial for any two polynomials $\mathbf{u}(x)$ and $\mathbf{v}(x)$ is $\mathbf{u}(x) \star \mathbf{v}(x) = \sum u_i v_i x^i$.

A binary *circulant* matrix M is a $p \times p$ matrix obtained by cyclically right shifting the first row:

$$M = \begin{pmatrix} m_0 & m_1 & \cdots & m_{p-1} \\ m_{p-1} & m_0 & \cdots & m_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ m_1 & m_2 & \cdots & m_0 \end{pmatrix}. \quad (1)$$

Thus any circulant matrix M is completely described by only its first row $\mathbf{m} = (m_0, \dots, m_{p-1})$. Note that a circulant matrix is also obtained by cyclically down shifting its first column. We shall see that the classical matrix operations of addition and multiplication preserve the circulant structure of matrices. It is possible to characterise the i -th row of a circulant matrix M as the polynomial:

$$x^i \cdot \mathbf{m}(x) \mod (x^p - 1).$$

If one looks at the product $\mathbf{b} \times M$ of a circulant matrix M with a binary vector $\mathbf{b} = (b_0, \dots, b_{p-1})$ then it exactly corresponds to the p -bit vector represented by the polynomial $\mathbf{b}(x) \cdot \mathbf{m}(x) \mod (x^p - 1)$. This property naturally extends to the product of two $p \times p$ circulant matrices M and N . Indeed, the first row of $M \times N$ is exactly $\mathbf{m}(x) \cdot \mathbf{n}(x) \mod (x^p - 1)$ and the i -th row of $M \times N$ is represented by the polynomial:

$$(x^i \cdot \mathbf{m}(x)) \cdot \mathbf{n}(x) \mod (x^p - 1) = x^i \cdot (\mathbf{m}(x) \cdot \mathbf{n}(x)) \mod (x^p - 1).$$

We have therefore the following result.

Proposition 1. *Let \mathfrak{C}_p be the set of binary $p \times p$ circulant matrices, then there exists an isomorphism between the rings $(\mathfrak{C}_p, +, \times)$ and $(\mathbb{F}_2[x]/(x^p - 1), +, \cdot)$:*

$$(\mathfrak{C}_p, +, \times) \simeq (\mathbb{F}_2[x]/(x^p - 1), +, \cdot)$$

Remark 1. *The first column of a circulant matrix M defined by $\mathbf{m}(x)$ corresponds to the polynomial $\mathbf{m}^*(x) = x^p \cdot \mathbf{m}(\frac{1}{x}) \mod (x^p - 1)$.*

Proposition 1 can be used to provide a simple characterisation of invertible matrices of circulant matrices:

Proposition 2. *A $p \times p$ circulant matrix M is invertible if and only if $\mathbf{m}(x)$ is prime with $x^p - 1$.*

Proof. One has only to prove that the invert of a circulant matrix M defined by a polynomial $\mathbf{m}(x)$ of $\mathbb{F}_2[x]/(x^p - 1)$ is necessarily a circulant matrix. Assume that there exists N such that $N \times M = M \times N = I_p$ with I_p being the $p \times p$ identity matrix. Let $\mathbf{n} = (n_0, \dots, n_{p-1})$ be the first row of N . We have previously seen that the product $\mathbf{n} \times M$ can be seen as the polynomial $\mathbf{n}(x) \cdot \mathbf{m}(x) \pmod{(x^p - 1)}$. This latter polynomial is equal to 1 by assumption. Consequently, for any i such that $0 \leq i \leq p - 1$ we also have $(x^i \cdot \mathbf{n}(x)) \cdot \mathbf{m}(x) = x^i \pmod{(x^p - 1)}$ which proves that the circulant matrix defined by $\mathbf{n}(x)$ is the invert of M . Therefore N is circulant. \square

A matrix G of size $k \times n$ is *p-block circulant* with $k = k_0 p$ and $n = n_0 p$ where k_0 and n_0 are positive integers if there exist $p \times p$ circulant matrices $G_{i,j} \in \mathbb{C}_p$ such that:

$$G = \begin{pmatrix} G_{1,1} & \cdots & G_{1,n_0} \\ \vdots & & \vdots \\ G_{k_0,1} & \cdots & G_{k_0,n_0} \end{pmatrix}$$

It is straightforward to see that the set of block circulant matrices is stable by matrix addition and matrix multiplication. It is therefore natural to establish an identification between a block circulant matrix G with a polynomial $k_0 \times n_0$ matrix $\mathbf{G}(x)$ with entries in $\mathbb{F}_2[x]/(x^p - 1)$ by means of the mapping that sends each block $G_{i,j}$ to the polynomial $\mathbf{g}_{i,j}(x)$ defining it.

Proposition 3. Let \mathfrak{B}_{k_0,n_0}^p be the set of p -block circulant matrices of size $k_0 \times n_0$. Let $R_p = \mathbb{F}_2[x]/(x^p - 1)$ and define by $\mathfrak{M}_{k_0,n_0}(R_p)$ the set of $k_0 \times n_0$ matrices with coefficients in R_p . There exists a ring isomorphism between \mathfrak{B}_{k_0,n_0}^p and $\mathfrak{M}_{k_0,n_0}(R_p)$:

$$\begin{aligned} \mathfrak{B}_{k_0,n_0}^p &\simeq \mathfrak{M}_{k_0,n_0}(R_p) \\ G &\longmapsto \mathbf{G}(x). \end{aligned}$$

In particular any p -block circulant matrix G is invertible if and only if $\det(\mathbf{G})(x)$ is prime with $x^p - 1$ and its inverse is also a p -block circulant matrix.

2.2. Cyclic and Quasi-Cyclic Codes

A (binary) linear code \mathcal{C} of length n and dimension k is a k -dimensional vector subspace of \mathbb{F}_2^n . The elements of a code are called *codewords*. A *generator matrix* G of \mathcal{C} is a $k' \times n$ matrix with $k' \geq k$ whose rows generate \mathcal{C} . A *parity-check matrix* H of \mathcal{C} is an $r \times n$ matrix with $r \geq n - k$ such that for any codeword $\mathbf{c} \in \mathcal{C}$ we have:

$$H \times \mathbf{c}^T = 0.$$

It is well-known that if a generator matrix of \mathcal{C} is of the form $(I|A)$ where I is the identity matrix then $(A^T|I)$ is a parity-check matrix for \mathcal{C} . Such a generator matrix is said to be *in reduced echelon form*. A code \mathcal{C}' is said to be *permutation equivalent* to \mathcal{C} if there exists a permutation of the symmetric group of order n that

reorders the coordinates of codewords of \mathcal{C}' into codewords of \mathcal{C} . It is convenient to consider equivalent codes as the same code.

A *cyclic* code \mathcal{C} of length n is an ideal of the ring $\mathbb{F}_2[x]/(x^n - 1)$. Such a code is characterised by a unique polynomial $\mathbf{g}(x)$ divisor of $(x^n - 1)$. Let r be the degree of $\mathbf{g}(x)$. Any codeword $\mathbf{c}(x)$ is obtained as a product in $\mathbb{F}_2[x]$ of the form:

$$\mathbf{c}(x) = \mathbf{m}(x) \cdot \mathbf{g}(x)$$

where $\mathbf{m}(x)$ is a polynomial of $\mathbb{F}_2[x]$ of degree $n - 1 - r$. \mathcal{C} is a linear code of dimension $k = n - r$. The polynomial $\mathbf{g}(x)$ is called the *generator polynomial* of the cyclic code \mathcal{C} and we shall write $\mathcal{C} = \langle \mathbf{g}(x) \rangle$.

A code \mathcal{C} is *quasi-cyclic of index p* if there exists a generator matrix G that is p -block circulant. We assume that all the $G_{i,j}$'s are square matrices of size $p \times p$ and therefore $n = n_0 p$ and $k = k_0 p$. Cyclic codes of length n are thus quasi-cyclic codes of index n where a generator matrix is a circulant matrix associated to its generator polynomial.

A useful method developed in [11] for obtaining quasi-cyclic codes of length $n = pn_0$ and index p is to consider a cyclic code \mathcal{C} generated by a polynomial $\mathbf{g}(x)$ and construct the subcode $S_{n_0}(\mathbf{c})$ spanned by a codeword $\mathbf{c}(x)$ and its $p - 1$ shifts modulo $(x^n - 1)$ of n_0 bits $x^{n_0} \cdot \mathbf{c}(x), \dots, x^{(p-1)n_0} \cdot \mathbf{c}(x)$. However note that $S_{n_0}(\mathbf{c})$ does not admit a p -block circulant generator matrix. Actually, one has to consider the equivalent code of \mathcal{C} obtained with the permutation π that maps any $an_0 + b$ to $bp + a$ with $1 \leq a \leq p - 1$ and $0 \leq b \leq n_0 - 1$. It means that up to a permutation any codeword $\mathbf{c}(x)$ of a cyclic code \mathcal{C} can be seen as a vector $\mathbf{c} = (\mathbf{c}_0, \dots, \mathbf{c}_{n_0-1})$ where each \mathbf{c}_i belongs to $\mathbb{F}_2^p \simeq \mathbb{F}_2[x]/(x^p - 1)$ and such that the vector $\mathbf{c}' = (\mathbf{c}'_0, \dots, \mathbf{c}'_{n_0-1})$ with $\mathbf{c}'_j(x) = x \cdot \mathbf{c}_j(x) \bmod (x^p - 1)$ is also a codeword of $S_{n_0}(\mathbf{c})$.

3. A McEliece Cryptosystem Based on Subcodes of a BCH Code

3.1. Description

Let \mathcal{C}_0 be a cyclic code of length $n = pn_0$ and let k be the dimension of \mathcal{C}_0 . Assume that \mathcal{C}_0 admits an $k' \times n$ generator matrix with $k' \geq k$ and such that $k' = pk_0$. For simplicity, we set $k' = k$. Let $\mathbf{c}_1(x), \mathbf{c}_2(x), \dots, \mathbf{c}_{k_0-1}(x)$ be random codewords of \mathcal{C}_0 and consider the linear code \mathcal{C} defined as:

$$\mathcal{C} = S_{n_0}(\mathbf{c}_1) + \dots + S_{n_0}(\mathbf{c}_{k_0-1}).$$

We assume that \mathcal{C} is of dimension $k - p = p(k_0 - 1)$. Recall from Section 2.2 that up to a permutation any n -bit vector $\mathbf{c}_i(x)$ with $1 \leq i \leq k_0 - 1$ can be seen as a vector $(\mathbf{c}_{i,0}, \dots, \mathbf{c}_{i,n_0-1})$ where each $\mathbf{c}_{i,j}$ can also be seen as an element of $\mathbb{F}_2[x]/(x^p - 1)$. Thus \mathcal{C} is a quasi-cyclic code of index p whose generator matrix

$\mathbf{G}(x)$ in p -block circulant form is:

$$\mathbf{G}(x) = \begin{pmatrix} \mathbf{c}_{1,1}(x) & \cdots & \mathbf{c}_{1,n_0}(x) \\ \vdots & & \vdots \\ \mathbf{c}_{k_0-1,1}(x) & \cdots & \mathbf{c}_{k_0-1,n_0}(x) \end{pmatrix}.$$

The variant of the McEliece cryptosystem proposed in [11] starts from a secret subcode \mathcal{C} of dimension $p(k_0 - 1)$ of a primitive BCH code \mathcal{C}_0 obtained by the method explained above. A secret permutation π of the symmetric group of order n_0 hides the structure of \mathcal{C} while keeping its quasi-cyclic structure by publicly making available a generator matrix $\mathbf{G}^\pi(x)$ defined by:

$$\mathbf{G}^\pi(x) = \begin{pmatrix} \mathbf{c}_{1,\pi(1)}(x) & \cdots & \mathbf{c}_{1,\pi(n_0)}(x) \\ \vdots & & \vdots \\ \mathbf{c}_{k_0-1,\pi(1)}(x) & \cdots & \mathbf{c}_{k_0-1,\pi(n_0)}(x) \end{pmatrix}.$$

The cyclic code \mathcal{C}_0 given in [11] is a primitive BCH of length $2^m - 1$ and dimension $n - tm$ where t is a positive integer. Two sets of parameters are proposed respectively corresponding to 2^{100} and 2^{80} security levels.

- Parameters A: $m = 12$, $t = 26$, $p = 91$, $n_0 = 45$, and $k_0 = 43$.
- Parameters B: $m = 11$, $t = 31$, $p = 89$, $n_0 = 23$ and $k_0 = 21$.

Note that we always have $p > n_0$. This property will be useful for cryptanalyzing the cryptosystem.

3.2. Structural Cryptanalysis

We describe a method that recovers the secret permutation π of the cryptosystem of [11] and thus reveals the secret key of any user. It exploits three facts:

1. The code \mathcal{C}_0 admits a binary $(n - k) \times n$ parity check matrix H_0 which can be assumed to be known. There are only a few different primitive BCH codes for a given parameter set (n, m, t) and we can try all of them. This is a consequence of the fact that the number of such codes is clearly upper-bounded by the number of primitive polynomials of degree m . For instance for the parameter set B, this number is equal to 176.
2. Since \mathcal{C} is a subcode of \mathcal{C}_0 , any n -bit codeword \mathbf{c} of \mathcal{C} must satisfy the equation:

$$H_0 \times \mathbf{c}^T = 0. \quad (2)$$

3. Permuting through a permutation π the columns of a polynomial generator matrix $\mathbf{G}(x)$ of \mathcal{C} can also be translated into a matrix product by the associated $n_0 \times n_0$ permutation matrix $\mathbf{\Pi}$ of π . Note that $\mathbf{\Pi}$ can also be seen as a polynomial matrix $\mathbf{\Pi}(x) \in \mathfrak{B}_{n_0, n_0}^p$ where 0 (resp. 1) entry corresponds to 0 (resp. 1) constant polynomial so that we have:

$$\mathbf{G}^\pi(x) = \mathbf{G}(x) \times \mathbf{\Pi}(x). \quad (3)$$

Note that Equation (3) can be rewritten as an equality between binary p -block circulant matrices:

$$G^\pi = G \times \Pi, \quad (4)$$

where G^π is the $(k-p) \times n$ public generator matrix and $\Pi = \mathbf{\Pi} \otimes I_p$ with I_p being the $p \times p$ identity matrix. Finding $\mathbf{\Pi}$ actually amounts to solve a linear system of n_0^2 unknowns representing the entries of $\mathbf{\Pi}^{-1}$ such that:

$$H_0 \times (G^\pi \times \Pi^{-1})^T = 0. \quad (5)$$

In other words, each row of the public matrix G^π after being permuted by Π^{-1} must satisfy Equation (2). This is a linear system since Π^{-1} may be rewritten as $\mathbf{\Pi}^{-1} \otimes I_p$. This means that each row of G^π provides $(n-k)$ binary linear equations verified by Π^{-1} . Thus Equation (5) gives a total number of $(k-p)(n-k)$ linear equations that must be satisfied by n_0^2 unknowns.

The cryptanalysis of [11] amounts to solve an over-constrained linear system constituted of $p^2(k_0-1)(n_0-k_0)$ equations and n_0^2 unknowns since as we have remarked that $p > n_0$. For instance, Parameters B give 529 unknowns that should satisfy 316,840 equations. As for Parameters A we obtain 2,025 unknowns that satisfy 695,604 equations. Many of these equations are obviously linearly dependent. The success of this method heavily depends on the size of the solution vector space. An implementation in Magma software actually always gave in both cases a vector space of dimension one. This revealed the secret permutation.

4. A Cryptosystem Based on Quasi-Cyclic LDPC Codes

4.1. Description

LDPC codes are linear codes defined by sparse binary parity-check matrices. We assume as in [1] that $n = pn_0$ and $k = p(n_0 - 1)$, and we consider a parity-check matrix H of the following form:

$$H = (\begin{array}{cccc} H_1 & \cdots & H_{n_0} \end{array}) \quad (6)$$

where each matrix H_j is a sparse circulant matrix of size $p \times p$. Without loss of generality, H_{n_0} is chosen to have full rank. Each column of H has a fixed weight d_v which is very small compared to the length n . We also assume that one has a good approximation of the number t of correctable errors through iterative decoding of the code defined by H .

The quasi-cyclic LDPC cryptosystem proposed in [1] takes two invertible p -block circulant matrices S and Q of size $k \times k$ and $n \times n$ respectively. The matrix S (resp. Q) is chosen such that the weight of each row and each column is s (resp. m). The private key consists of the parity-check matrix H and the matrices S and Q . In order to produce the public key, one has to compute a generator matrix G' in reduced echelon form and make public the matrix $G = S^{-1} \times G' \times Q^{-1}$. The plaintext space is the set \mathbb{F}_2^k and the ciphertext space is \mathbb{F}_2^n . If one wishes to encrypt a message $\mathbf{x} \in \mathbb{F}_2^k$, one has to randomly choose a n -bit vector \mathbf{e} of weight

$t' \leq t/m$ and compute $\mathbf{c} = \mathbf{x} \times G + \mathbf{e}$. The decryption step consists in iteratively decoding $\mathbf{c} \times Q = \mathbf{x} \times S^{-1} \times G' + \mathbf{e} \times Q$ to output $\mathbf{z} = \mathbf{x} \times S^{-1}$ and then computing $\mathbf{x} = \mathbf{z} \times S$. The crucial point that makes this cryptosystem valid is that $\mathbf{e} \times Q$ is a correctable error because its weight is less than or equal to $t'm$.

4.2. Some Remarks on the Choice of the Parameters

The authors suggest to take a matrix Q in diagonal form. They also suggest the following values: $p = 4032$, $n_0 = 4$, $d_v = 13$, $m = 7$ and $t = 190$ ($t' = 27$). Finally, each block circulant matrix of S has a column/row weight equals to m so as to have $s = m(n_0 - 1)$. Unfortunately, for this specific constraint, there is a flaw in this choice because the matrix S is not invertible. This follows from the fact that in this case $x - 1$ always divides $\det(\mathbf{S})(x)$ which is therefore not coprime with $x^p - 1$ and this implies that $\mathbf{S}(x)$ is not invertible. This can be proved by using the following arguments.

Lemma 1. *Let $\mathbf{S}(x) = (\mathbf{s}_{i,j}(x))$ in $\mathfrak{M}_{n_0-1, n_0-1}(R_p)$ and define the binary matrix $\tilde{S} = (\tilde{s}_{i,j})$ by $\tilde{s}_{i,j} = \text{wt}(\mathbf{s}_{i,j}) \pmod{2}$. We have then:*

$$\det(\tilde{S}) = \text{wt}(\det(\mathbf{S})) \pmod{2}.$$

Proof. This comes from the fact that $\text{wt}(\mathbf{u} + \mathbf{v}) = \text{wt}(\mathbf{u}) + \text{wt}(\mathbf{v}) - 2\text{wt}(\mathbf{u} \star \mathbf{v})$ for any $\mathbf{u}(x)$ and $\mathbf{v}(x)$ in $\mathbb{F}_2[x]$ which implies that:

$$\begin{cases} \text{wt}(\mathbf{u} + \mathbf{v}) &= \text{wt}(\mathbf{u}) + \text{wt}(\mathbf{v}) \pmod{2} \\ \text{wt}(\mathbf{u} \cdot \mathbf{v}) &= \text{wt}(\mathbf{u}) \cdot \text{wt}(\mathbf{v}) \pmod{2}. \end{cases}$$

□

Proposition 4. *For any $\mathbf{S}(x)$ in $\mathfrak{M}_{3,3}(R_p)$ such that each $\mathbf{s}_{i,j}$ is of weight m then $x - 1$ divides $\det(\mathbf{S})(x)$.*

Proof. By using the same notation as in the previous lemma we know that $\det(\tilde{S})$ is equal to zero since \tilde{S} is the all one matrix. From the previous lemma it follows that $\det(\mathbf{S})(x)$ has a support of even weight. This implies that $x - 1$ divides $\det(\mathbf{S})(x)$. □

In order to avoid this situation we introduce as few polynomials of weight different from m in \mathbf{S} such that $\det(\tilde{S}) = 1$. A possible choice is the following one. First we choose a nonsingular \tilde{S} equal to

$$\tilde{S} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

When $\tilde{s}_{ij} = 1$ we choose the corresponding entry $\mathbf{s}_{ij}(x)$ to be of weight m and if $\tilde{s}_{ij} = 0$ we choose the corresponding entry $\mathbf{s}_{ij}(x)$ to be of weight $m - 1$.

It should also be mentioned that a decoding attack searching for a word of weight less than $t = 27$ in a code of length $n = 16128$ and dimension $k = 12096$ as proposed by using the algorithm given in [6] has a work factor of about $2^{78.5}$. Note that this work factor may even be decreased with the algorithm of [7].

4.3. Structural Attack

4.3.1. Preliminaries. The goal of this attack is to recover the secret code \mathcal{C} defined by the parity-check matrix H given in Equation (6). We know that S and Q are equivalently defined by polynomials $\mathbf{s}_{i,j}(x)$ and $\mathbf{q}_{i,j}(x)$ respectively. Q is chosen to be in diagonal form, that is to say $\mathbf{q}_{i,j}(x) = 0$ if $i \neq j$. For the sake of simplicity, we set $\mathbf{q}_i(x) = \mathbf{q}_{i,i}(x)$. Moreover the polynomials $\mathbf{q}_i(x)$ are invertible modulo $x^p - 1$ since Q is invertible. It is also straightforward to remark that the secret generator matrix G' is equal to:

$$G' = \left(\begin{array}{c|c} & \begin{matrix} (H_{n_0}^{-1}H_1)^T \\ \vdots \\ (H_{n_0}^{-1}H_{n_0-1})^T \end{matrix} \\ \hline I_k & \end{array} \right).$$

In others words, if we denote by $G_{\leq k}$ the matrix obtained by taking the k first columns of G then we have:

$$G_{\leq k} = S^{-1} \times \begin{pmatrix} Q_1^{-1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & Q_{n_0-1}^{-1} \end{pmatrix}.$$

This implies that $G_{\leq k}^{-1}$ is a p -block circulant matrix defined by polynomials $\mathbf{g}_{i,j}(x)$ that satisfies the following equations:

$$\mathbf{g}_{i,j}(x) = \mathbf{q}_i(x) \cdot \mathbf{s}_{i,j}(x) \mod (x^p - 1). \quad (7)$$

Note that the weight of $\mathbf{g}_{i,j}(x)$ is at most m^2 . Actually, due the fact that the secret polynomials have very low weights, we shall see that the support of $\mathbf{g}_{i,j}(x)$ is exactly m^2 with a good probability. For the sake of simplicity, we set $\mathbf{q}_i(x) = x^{e_1} + \cdots + x^{e_m}$ and $\mathbf{s}_{i,j}(x) = x^{\ell_1} + \cdots + x^{\ell_m}$ with $0 \leq e_a \leq p-1$ and $0 \leq \ell_a \leq p-1$ for any $1 \leq a \leq m$. We fix $\mathbf{q}_i(x)$ and we assume that the monomials x^{ℓ_a} of $\mathbf{s}_{i,j}(x)$ are independently and uniformly chosen. We wish to estimate the probability that the support of $\mathbf{g}_{i,j}(x)$ contains the support of at least one shift $x^{\ell_a} \cdot \mathbf{q}_i(x)$, and the probability that the weight of $\mathbf{g}_{i,j}(x)$ is exactly m^2 .

Lemma 2. *Let ℓ_1, \dots, ℓ_w be w different integers such that $0 \leq \ell_a \leq p-1$ for $1 \leq a \leq w$. For any random integer $0 \leq \ell \leq p-1$ such that ℓ is different from ℓ_1, \dots, ℓ_w , we have:*

$$\Pr \{ (x^{\ell_1} + \cdots + x^{\ell_w}) \cdot \mathbf{q}_i(x) \star x^\ell \cdot \mathbf{q}_i(x) \neq 0 \} \leq w \frac{m(m-1)}{p-w}$$

Proof. Set first $\mathbf{r}(x) = (x^{\ell_1} + \cdots + x^{\ell_w}) \cdot \mathbf{q}_i(x)$. By the union bound we have:

$$\Pr \{ \mathbf{r}(x) \star x^\ell \cdot \mathbf{q}_i(x) \neq 0 \} \leq \sum_{a=1}^w \Pr \{ x^{\ell_a} \cdot \mathbf{q}_i(x) \star x^\ell \cdot \mathbf{q}_i(x) \neq 0 \}$$

The probability $\Pr\{x^{\ell_a} \cdot \mathbf{q}_i(x) \star x^\ell \cdot \mathbf{q}_i(x) \neq 0\}$ is at most the fraction of integers ℓ different from ℓ_1, \dots, ℓ_w such that there exist $1 \leq b \leq m$ and $1 \leq c \leq m$ with:

$$\ell_a + e_b = \ell + e_c \pmod{p}.$$

Thus, this fraction is given by the ratio of the number of pairs (e_b, e_c) with $b \neq c$ to the number of possible values for ℓ which is exactly $m(m-1)/(p-w)$. \square

Proposition 5. *The probability $\Pr\{x^\ell \cdot \mathbf{q}_i(x) \subset \mathbf{g}_{i,j}(x)\}$ for ℓ in $\{\ell_1, \dots, \ell_m\}$ that the support of $\mathbf{g}_{i,j}(x)$ contains the support of $x^\ell \cdot \mathbf{q}_i(x)$ is lower-bounded by:*

$$\Pr\{x^\ell \cdot \mathbf{q}_i(x) \subset \mathbf{g}_{i,j}(x)\} \geq \left(1 - \frac{m(m-1)}{p-1}\right)^{m-1}.$$

Proof. This inequality is obtained by taking $w = 1$ in Lemma 2 and by the independence of the choice of the $(m-1)$ other monomials of $\mathbf{s}_{i,j}(x)$. \square

Proposition 6. *The probability q that $\mathbf{g}_{i,j}(x)$ is exactly of weight m^2 is lower-bounded by:*

$$q \geq \prod_{w=1}^{m-1} \left(1 - w \cdot \frac{m(m-1)}{p-w}\right).$$

Proof. For any $2 \leq w \leq m$, let E_w denote the event that

$$E_w : (x^{\ell_1} + \dots + x^{\ell_{w-1}}) \cdot \mathbf{q}_i(x) \star x^{\ell_w} \cdot \mathbf{q}_i(x) = 0$$

when each monomial x^{ℓ_a} is uniformly and independently chosen. We also set E_1 as the whole universe. Then we have:

$$q \geq \Pr\{E_2 \cap \dots \cap E_m\}$$

Using Bayes' rule we also have

$$\Pr\{E_2 \cap \dots \cap E_m\} = \prod_{w=1}^m \Pr\{E_w | E_{w-1} \cap \dots \cap E_1\}.$$

But by Lemma 2 we know that $\Pr\{E_w | E_{w-1} \cap \dots \cap E_1\} \geq \left(1 - w \cdot \frac{m(m-1)}{p-w}\right)$. \square

4.3.2. Different Strategies.

First Strategy. We have seen in Lemma 2 that the support of $\mathbf{g}_{i,j}(x)$ contains with very high probability the support of at least¹ a shifted version of $\mathbf{q}_i(x)$ since for the parameters given in [1], we obtain $\Pr\{x^\ell \cdot \mathbf{q}_i(x) \subset \mathbf{g}_{i,j}(x)\} \geq 0.94$. One possible strategy to recover the polynomial $\mathbf{q}_i(x)$ consists in enumerating m -tuples u_1, \dots, u_m that belong in the support of $\mathbf{g}_{i,j}(x)$ in order to form $\mathbf{u}(x) = \sum_a x^{u_a}$ such that $\mathbf{u}^{-1}(x) \cdot \mathbf{g}_{i,j'}(x)$ is of weight m for $1 \leq j' \leq n_0 - 1$. The cost of this attack is $O\left(\binom{m^2}{m} \cdot p^2\right)$ which corresponds to $2^{50.3}$ operations for the specific parameters proposed.

¹Actually, the support of $\mathbf{g}_{i,j}(x)$ contains with good probability all the supports of $x^{\ell_a} \cdot \mathbf{q}_i(x)$ with $1 \leq a \leq m$ since $q \geq 0.79$ for the proposed parameters.

Second Strategy. We present another strategy that can be used to recover secret matrices S and thus matrices Q_1, \dots, Q_{n_0-1} . This strategy requires to search for codewords of very low weight in a linear code. The most efficient algorithm that accomplishes this task is the algorithm of [3] which improves upon Stern's algorithm [23]. However in order to derive a simple bound on the time complexity, we consider this second algorithm as in [1]. The work factor $\Omega_{n,k,w}$ of Stern's algorithm to find A_w codewords of weight w in a code of length n and dimension k satisfies $\Omega_{n,k,w} \geq \frac{N}{A_w P_w}$ where (g, ℓ) are two parameters and N is the number of binary operations required for each iteration

$$N = (n-k)^3/2 + k(n-k)^2 + 2g\ell \binom{k/2}{g} + 2g(n-k) \frac{\binom{k/2}{g}^2}{2^\ell}. \quad (8)$$

P_w represents the probability of finding a given codeword of weight w

$$P_w = \frac{\binom{w}{g} \binom{n-w}{k/2-g}}{\binom{n}{k/2}} \frac{\binom{w-g}{g} \binom{n-k/2-w+g}{k/2-g}}{\binom{n-k/2}{k/2}} \frac{\binom{n-k-w+2g}{\ell}}{\binom{n-k}{\ell}}.$$

Recall that $G_{\leq k}^{-1}$ is specified by polynomials $\mathbf{g}_{i,j}(x)$. Let $\mathbf{d}_{i,j}(x)$ be the polynomial $\mathbf{g}_{i,j}(x) \cdot \mathbf{g}_{i,1}^{-1}(x) \pmod{(x^p - 1)}$ and consider the code \mathcal{E}_i defined by the following generator matrix:

$$E_i = \begin{pmatrix} I_p & D_{i,2} & \cdots & D_{i,n_0-1} \end{pmatrix}$$

where as usual the circulant matrix $D_{i,j}$ is characterised by the polynomial $\mathbf{d}_{i,j}(x)$. Then \mathcal{E}_i contains at least p codewords of low weight $(n_0 - 1)m = 21$ since

$$S_{i,1} \times E_i = \begin{pmatrix} S_{i,1} & S_{i,2} & \cdots & S_{i,n_0-1} \end{pmatrix}.$$

It is therefore possible to recover matrices $S_{i,1}, \dots, S_{i,n_0-1}$ with a complexity of 2^{32} operations by applying Stern's algorithm with $(g, \ell) = (3, 43)$ in order to find a codeword of weight 21 in a code of dimension p and length $(n_0 - 1)p = 12096$.

4.3.3. Extraction of the Secret Code. After recovering S, Q_1, \dots, Q_{n_0-1} , one is therefore able to compute the following generator matrix \tilde{G} defined by:

$$\tilde{G} = G' \times \begin{pmatrix} I_p & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & I_p & 0 \\ 0 & \cdots & 0 & Q_{n_0}^{-1} \end{pmatrix} = \left(I_k \left| \begin{array}{c} A_1 \\ \vdots \\ A_{n_0-1} \end{array} \right. \right)$$

where for $1 \leq i \leq n_0 - 1$, we set $A_i = (H_{n_0}^{-1} \times H_i)^T \times Q_{n_0}^{-1}$. Recall that matrices H_1, \dots, H_{n_0} and Q_{n_0} are still unknown. However, one can easily check that for any different i and j , we also have $(A_i \times A_j^{-1})^T = H_i \times H_j^{-1}$ whenever H_j is invertible. Thus, if we set $B_{i,j} = (A_i \times A_j^{-1})^T$ then for a fixed $1 \leq i \leq n_0 - 1$ and for any different integers j and j' , we have that $H_j \times B_{i,j} = H_{j'} \times B_{i,j'} = H_i$. Consider now the code defined by the following generator matrix G_1 :

$$G_1 = \begin{pmatrix} I_p & B_{2,1} & \cdots & B_{n_0-1,1} \end{pmatrix}.$$

It is easy to see that $H_1 \times G_1 = (H_1 \ H_2 \ \cdots \ H_{n_0-1})$. This also means that G_1 spans a code with a minimum distance that is smaller than $(n_0 - 1)d_v$. Therefore, by applying dedicated algorithms ([8] or [20, Volume I, Chapter 7]) searching for codewords of small weight, it is possible to recover matrices H_1, \dots, H_{n_0-1} . For instance, the work factor of Stern's algorithm for searching codewords of weight $(n_0 - 1)d_v = 3 * 13 = 39$ in a code of dimension $p = 4032$ and length $p(n_0 - 1) = 12096$ is about 2^{37} operations with $(g, \ell) = (3, 43)$.

Finally, we are able to compute $(H_i^T)^{-1} \times A_i = (H_{n_0}^{-1})^T \times Q_{n_0}^{-1}$ for any $1 \leq i \leq n_0 - 1$. Inverting this matrix and applying again the second strategy presented in Section 4.3.2, it is possible to find the matrices H_{n_0} and Q_{n_0} .

4.4. Example

We illustrate the previously described attacks with some randomly generated polynomials $s_{i,j}(x)$ and $q_{i,j}(x)$ of weight $m = 7$ and degree less than $p = 4032$ as given in [1]. We only put the exponents of the monomials that intervene in the expression of the polynomials. Recall that some coefficients $s_{i,j}(x)$ has to be of even weight (actually of weight $m - 1 = 6$) in order to generate an invertible matrix S . We implemented the attack in MAGMA software [4]. The running time on a Pentium 4 (2.80GHz) with 500 Mbytes RAM for the second strategy is 140 seconds. The last step that consists in recovering the secret LDPC code is performed by applying Canteaut-Chabaud algorithm. The work factor of this operation is about 2^{36} operations. Our implementation in MAGMA software finds a codeword of weight $(n_0 - 1)d_v = 39$ in about 15 minutes.

$$\begin{aligned}
H_1 &= [213, 457, 1467, 1702, 1786, 2015, 2155, 2197, 2569, 2744, 2823, 2902, 3710] \\
H_2 &= [6, 626, 868, 1102, 1564, 1894, 2401, 2595, 2982, 3570, 3605, 3771, 3835] \\
H_3 &= [615, 639, 1198, 1513, 1712, 1850, 1941, 2397, 2553, 3074, 3373, 3798, 3960] \\
H_4 &= [135, 149, 241, 735, 1265, 2075, 2869, 3111, 3218, 3625, 3760, 3785, 3969] \\
\\
S_{1,1} &= [24, 274, 334, 2025, 2574, 2661, 3601] \\
S_{1,2} &= [512, 1177, 2524, 2526, 2904, 2968, 3340] \\
S_{1,3} &= [930, 1175, 1210, 1459, 2200, 2303, 2811] \\
S_{2,1} &= [503, 1258, 1632, 1658, 2055, 2221, 2764] \\
S_{2,2} &= [989, 1256, 2568, 2625, 2906, 3139] \\
S_{2,3} &= [561, 616, 2499, 2787, 2835, 3061, 3865] \\
S_{3,1} &= [177, 465, 1659, 1958, 2795, 3605] \\
S_{3,2} &= [419, 461, 1540, 2262, 2435, 3474, 3587] \\
S_{3,3} &= [554, 1119, 1307, 2018, 2193, 2631, 3755] \\
\\
Q_1 &= [456, 578, 1551, 1562, 1992, 2919, 3476] \\
Q_2 &= [250, 268, 897, 1782, 2127, 3163, 3378] \\
Q_3 &= [14, 1132, 1672, 1716, 2164, 2723, 3409] \\
Q_4 &= [443, 593, 2401, 2615, 2981, 3612, 3993]
\end{aligned}$$

5. Conclusion

The idea to introduce quasi-cyclic codes and quasi-cyclic low density parity-check codes is motivated by practical concerns to reduce key sizes of McEliece cryptosystem. The first variant of [11] uses quasi-cyclic codes obtained from subcodes of a cyclic BCH code. The other variant of [1] uses quasi-cyclic low density parity-check codes. However, we have shown here that the cost of these two attempts at reducing key size is made at the expense of the security. Indeed, we have presented different structural cryptanalysis of these two variants of McEliece cryptosystem. The first attack is applied to the variant of [11] and extracts the secret permutation supposed to hide the structure of the secret codes. We show that the secret key recovery amounts to solve an over-constrained linear system. The second attack accomplishes a total break of [1]. In the first phase, we look for divisors of low weight of a given public polynomial. The last phase recovers the secret parity check matrix of the secret quasi-cyclic low density parity-check code by looking for low weight codewords in a punctured version of the secret code. An implementation shows that the first phase can be accomplished in about 140 seconds and the second phase in about 15 minutes.

However these results cannot be applied to the original McEliece's scheme using Goppa codes which represents up to now the only unbroken scheme. An open problem which would be desirable to solve is to come up with a way of reducing significantly the key sizes in this type of public-key cryptosystem by maintaining the security intact.

References

- [1] M. Baldi and G. F. Chiaraluce. Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes. In *IEEE International Symposium on Information Theory*, pages 2591–2595, Nice, France, March 2007.
- [2] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.
- [3] D.J. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. *PQCrypto*, pages 31–46, 2008.
- [4] W. Bosma, J. J. Cannon, and C. Playoust. The Magma algebra system I: The user language. *J. Symb. Comput.*, 24(3/4):235–265, 1997.
- [5] A. Shokrollahi C. Monico, J. Rosenthal. Using low density parity check codes in the McEliece cryptosystem. In *IEEE International Symposium on Information Theory (ISIT 2000)*, page 215, Sorrento, Italy, 2000.
- [6] A. Canteaut and H. Chabanne. A further improvement of the work factor in an attempt at breaking McEliece's cryptosystem. In *EUROCODE 94*, pages 169–173. INRIA, 1994.
- [7] A. Canteaut and F. Chabaud. Improvements of the attacks on cryptosystems based on error-correcting codes. Technical Report 95–21, INRIA, 1995.

- [8] A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.
- [9] P.L. Cayrel, A. Otmani, and D. Vergnaud. On Kabatianskii-Krouk-Smeets Signatures. In *Proceedings of the first International Workshop on the Arithmetic of Finite Fields (WAIFI 2007)*, Springer Verlag Lecture Notes, pages 237–251, Madrid, Spain, June 21–22 2007.
- [10] D. Engelbert, R. Overbeck, and A. Schmidt. A summary of McEliece-type cryptosystems and their security. In *Journal of Mathematical Cryptology*, volume 1, pages 151–199, 2007.
- [11] P. Gaborit. Shorter keys for code based cryptography. In *Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005)*, pages 81–91, Bergen, Norway, March 2005.
- [12] P. Gaborit and M. Girault. Lightweight code-based authentication and signature. In *IEEE International Symposium on Information Theory (ISIT 2007)*, pages 191–195, Nice, France, March 2007.
- [13] P. Gaborit, C. Lauradoux, and N. Sendrier. Synd: a fast code-based stream cipher with a security reduction. In *IEEE International Symposium on Information Theory (ISIT 2007)*, pages 186–190, Nice, France, March 2007.
- [14] P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In *Advances in Cryptology - EUROCRYPT'88*, volume 330/1988 of *Lecture Notes in Computer Science*, pages 275–280. Springer, 1988.
- [15] J. S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transactions on Information Theory*, 34(5):1354–1359, 1988.
- [16] Y. X. Li, R. H. Deng, and X.-M. Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–273, 1994.
- [17] R. J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.
- [18] L. Minder and A. Shokrollahi. Cryptanalysis of the Sidelnikov cryptosystem. In *Eurocrypt 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 347–360, Barcelona, Spain, 2007.
- [19] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems Control Inform. Theory*, 15(2):159–166, 1986.
- [20] V.S. Pless and W.C. Huffman, editors. *Handbook of coding theory*. North Holland, 1998.
- [21] V.M. Sidelnikov. A public-key cryptosystem based on binary Reed-Muller codes. *Discrete Mathematics and Applications*, 4(3), 1994.
- [22] V.M. Sidelnikov and S.O. Shestakov. On the insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 1(4):439–444, 1992.
- [23] J. Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 1988.
- [24] E. M. Gabidulin. *Public-Key Cryptosystems Based on Linear Codes*. 1995.

- [25] T.P. Berger and P. Loidreau. How to Mask the Structure of Codes for a Cryptographic Use. *Des. Codes Cryptography*, 35(1):63–79, 2005.
- [26] C. Wieschebrink. An Attack on a Modified Niederreiter Encryption Scheme. *Public Key Cryptography - PKC 2006*, Volume 3958/2006 of *Lecture Notes in Computer Science*, pages 14–26. Springer, 2006.
- [27] N. Sendrier. On the Concatenated Structure of a Linear Code. *AAECC*, 9(3):221–242, November 1998.

Ayoub Otmani

GREYC - Ensicaen - Université de Caen, Campus II, Boulevard Maréchal Juin, F-14050 Caen Cedex, France.

e-mail: Ayoub.Otmani@info.unicaen.fr

Jean-Pierre Tillich

INRIA, Projet Secret, BP 105, Domaine de Voluceau F-78153 Le Chesnay, France.

e-mail: jean-pierre.tillich@inria.fr

Léonard Dallot

GREYC - Ensicaen - Université de Caen, Campus II, Boulevard Maréchal Juin, F-14050 Caen Cedex, France.

e-mail: Leonard.Dallot@info.unicaen.fr